A Collective, Probabilistic Approach to Schema Mapping: Appendix

Angelika Kimmig
KU Leuven
angelika.kimmig@cs.kuleuven.be

Alex Memory
University of Maryland
memory@cs.umd.edu

Renée J. Miller
University of Toronto
miller@cs.toronto.edu

Lise Getoor
UC Santa Cruz
getoor@ucsc.edu

In this appendix we provide additional supplementary material to "A Collective, Probabilistic Approach to Schema Mapping" [1]. We include an additional extended example, supplementary experiment details, and proof for the complexity result stated in the main paper.

I. Example of Selection over ST TGDs

We extend the running example from the main paper to illustrate objective Eq. (9) of [1]. We use a reduced candidate set $\mathcal{C}' = \{\theta_1, \theta_3\}$ (Figure 1(d) in [1]) and the data in Figure 1(b)-(c) in [1], but omit the leader relation. A universal solution K_{θ_1} of I contains the task tuples (BigData, Bob, Null_1) and (ML, Alice, Null_2), while a K_{θ_3} contains the task tuples (BigData, Bob, Null_3) and (ML, Alice, Null_4) and the org tuples (Null_3, IBM) and (Null_4, SAP).

For θ_1, creates is 1 for tuple task(BigData, Bob, Null_1), and 0 for all other tuples, and covers is $2/3$ for task(ML, Alice, 111) and 0 otherwise. This is because task(ML, Alice, Null_2) partially explains the latter via a homomorphism mapping Null_2 to 111. Similarly, for θ_3, creates is 1 for task(BigData, Bob, Null_3) and org(Null_3,IBM), but 0 for task(ML, Alice, Null_4) and org(Null_4,SAP), which partially explain task(ML, Alice, 111) and org(111, SAP) to degree $3/3$ and $2/2$ respectively, via a homomorphism mapping Null_4 to 111, with corresponding values for covers. The different subsets of candidate st tgds thus obtain the following values for the individual parts and the total of objective function Eq. (9) of [1].

$\mathcal{M}$	$\sum 1 - \text{explains}$	$\sum \text{error}$	size	Eq. (9) of [1]
$\{\}$	4	0	0	4
$\{\theta_1\}$	$3\,1/3$	1	3	$7\,1/3$
$\{\theta_3\}$	2	2	4	8
$\{\theta_1, \theta_3\}$	2	3	7	12

As the data example is small compared to the mappings, the minimal value for the objective is that of the empty mapping, but we also see that $\{\theta_1\}$ is preferred over $\{\theta_3\}$, which in turn is preferred over $\{\theta_1, \theta_3\}$. The reason is that while θ_3 covers more tuples than θ_1, it also produces more errors and is larger. The fact that the empty mapping has a better objective value is an important guard against overfitting on too little data; this is easily overcome by slightly larger data instances. If we add at least five more projects X of the same kind as the ML one, i.e., pairs of tuples proj(X,N,1) and task(X,Alice,111), the preferred mapping is $\{\theta_3\}$, as the empty mapping cannot explain the new target tuples, θ_1 explains each to degree $2/3$, and θ_3 fully explains them (while no mapping introduces additional errors).

II. Scenario generation

We provide additional details of the scenario generation process discussed in Section VI-A of [1].

iBench. We used seven iBench primitives [2], [3]: CP copies a source relation to the target, changing its name. ADD copies a source relation and adds attributes; DL does the same, but removes attributes instead; and ADL adds and removes attributes to the same relation. The number that are added or removed are controlled by range parameters, which we set to (2,4). ME copies two relations, after joining them, to form a target relation. VP copies a source relation to form two, joined, target relations. VNM is the same as VP but introduces an additional target relation to form a N-to-M relationship between the other target relations.

Modifying the metadata evidence through random correspondences. If $\pi_{Corresp} > 0$ (cf. Table I of [1]), we introduce additional correspondences as follows. We randomly select $\pi_{Corresp}$ percent of the target relations. For every selected target relation T, we randomly select a source relation S from those of the iBench primitive invocations not involving T (so Clio [4] can generate $\mathcal{M}_G$ as part of $\mathcal{C}$). For each attribute of T, we introduce a correspondence to a randomly selected attribute of S.

Modifying the data instance. As certain errors and certain unexplained tuples can be removed prior to optimization (cf. Section III-C of [1]), we restrict data instance modifications to non-certain errors and non-certain unexplained tuples (with respect to $\mathcal{M}_G$). Note that in our scenarios, $\mathcal{M}_G \subseteq \mathcal{C}$, and thus $K_G \subseteq K_\mathcal{C}$. So each tuple in $K_\mathcal{C}$ is either generated by both $\mathcal{M}_G$ and $\mathcal{C}-\mathcal{M}_G$, only by $\mathcal{M}_G$ (i.e., a non-certain error tuple if deleted from J), or only by $\mathcal{C}-\mathcal{M}_G$ (i.e., a non-certain unexplained tuple if added to J). As tuples in $K_\mathcal{C}$ may have nulls, we take into account homomorphisms when determining which of these cases applies to a given tuple. We randomly select $\pi_{Unexplained}\%$ of the potential non-certain unexplained tuples, which we add to J, and $\pi_{Errors}\%$ of the potential non-certain error tuples, which we delete from J.

III. Mapping selection is NP-hard

We provide a proof for the complexity result stated in Section III-C of the main paper.

Theorem 1: The mapping selection problem for full st tgds as defined in Eq. (4) of [1] is NP-hard.

Proof: We use a reduction from SET COVER, which is well known to be NP-complete, and is defined as follows:

Given a finite set U, a finite collection $R = \{R_i \mid R_i \subseteq U, 1 \leq i \leq k\}$ and a natural number $n \leq k$, is there a set $R' \subseteq R$ consisting of at most n sets R_i such that $\bigcup_{R_i \in R'} R_i = U$?

We first consider the decision variant of mapping selection, which is defined as follows:

Given schemas $\mathbf{S}$, $\mathbf{T}$, a data example (I, J), a set $\mathcal{C}$ of candidate *full* st tgds, and a natural number m, is there a selection $\mathcal{M} \subseteq \mathcal{C}$ with $F(\mathcal{M}) \leq m$?

where $F(\mathcal{M})$ is the function minimized in Eq. (4) of [1], i.e.,

$$F(\mathcal{M}) = \sum_{t \in J} [1 - \mathsf{explains}_{\mathsf{full}}(\mathcal{M}, t)] + \sum_{t \in K_{\mathcal{C}} - J} [\mathsf{error}_{\mathsf{full}}(\mathcal{M}, t)] + \mathsf{size}_{\mathsf{m}}(\mathcal{M}) \quad (1)$$

We construct a mapping selection decision instance from a SET COVER instance as follows. We set $m = 2n$, introduce an auxiliary domain $D = \{1, \ldots, m+1\}$, and define

$$\begin{aligned} \mathbf{S} &= \{R_i/2 \mid R_i \in R\} \\ \mathbf{T} &= \{U/2\} \\ \mathcal{C} &= \{R_i(X,Y) \rightarrow U(X,Y) \mid R_i \in R\} \\ J &= \{U(x,y) \mid (x,y) \in U \times D\} \\ I &= \bigcup_{R_i \in R} \{R_i(x,y) \mid (x,y) \in R_i \times D\} \end{aligned}$$

It is easily verified that this construction is polynomial in the size of the SET COVER instance. We next show that the answers to SET COVER and the constructed mapping selection problem coincide.

For each R_i, the candidate st tgd $\theta_i = R_i(X,Y) \rightarrow U(X,Y)$ has size two, makes no errors (as $R_i \subseteq U$), and for each $x \in R_i$ explains the tuples $U(x,1), \ldots, U(x,m+1)$. We thus have

$$F(\mathcal{M}) = \sum_{t \in J} [1 - \mathsf{explains}_{\mathsf{full}}(\mathcal{M}, t)] + 2 \cdot |\mathcal{M}| \quad (2)$$

$$= (m+1) \cdot \left(|U| - |\bigcup_{\theta_i \in \mathcal{M}} R_i| \right) + 2 \cdot |\mathcal{M}| \quad (3)$$

A mapping $\mathcal{M} \subseteq \mathcal{C}$ with $F(\mathcal{M}) \leq m = 2n$ thus exists if and only if $|\bigcup_{\theta_i \in \mathcal{M}} R_i| = |U|$ and $|\mathcal{M}| \leq n$, which is exactly the case where $\mathcal{M}$ encodes a covering selection with at most n sets. Furthermore, if such mappings exist, the optimal mapping according to Eq. (4) of [1] is one of them, and a polynomial time solution for mapping selection with full st tgds can thus be used to find a candidate solution that can be verified or rejected in polynomial time to answer SET COVER. ■

We note that the mapping selection problem for arbitrary st tgds as defined in Eq. (9) of [1] coincides with the one in Eq. (4) of [1] if all candidates are full, and thus is NP-hard as well. Furthermore, the reduction used in the proof directly generalizes to the following weighted version of the optimization criterion:

$$F(\mathcal{M}) = w_1 \cdot \sum_{t \in J} [1 - \mathsf{explains}_{\mathsf{full}}(\mathcal{M}, t)] + w_2 \cdot \sum_{t \in K_{\mathcal{C}} - J} [\mathsf{error}_{\mathsf{full}}(\mathcal{M}, t)] + w_3 \cdot \sum_{\theta \in \mathcal{M}} \mathsf{size}(\theta)$$

with positive integer weights w_1, w_2, w_3 and any size function that assigns equal size to the candidate mappings $\theta_i = R_i(X,Y) \rightarrow U(X,Y)$. More precisely, setting $m = \mathsf{size}(\theta_1) \cdot w_3 \cdot n$ in the proof above shows that this generalization is NP-hard as well.

REFERENCES

[1] A. Kimmig, A. Memory, R. J. Miller, and L. Getoor, "A collective, probabilistic approach to schema mapping," in *ICDE*, (accepted) 2017.

[2] B. Alexe, W.-C. Tan, and Y. Velegrakis, "STBenchmark: towards a benchmark for mapping systems," *PVLDB*, vol. 1, no. 1, pp. 230–244, 2008.

[3] P. C. Arocena, B. Glavic, R. Ciucanu, and R. J. Miller, "The iBench Integration Metadata Generator," *PVLDB*, vol. 9, no. 3, pp. 108–119, 2015.

[4] R. Fagin, L. M. Haas, M. A. Hernández, R. J. Miller, L. Popa, and Y. Velegrakis, "Clio: Schema Mapping Creation and Data Exchange," in *Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos*, 2009, pp. 198–236.